Patent
Attorney's Docket No. 033275-223

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of	)	
	)	
GEORGES DEVAQUET, et al.	)	Group Art Unit: Unassigned
	)	
Application No.: Unassigned	)	Examiner: Unassigned
	)	
Filed: June 15, 2001	)	
	)	
For: METHOD FOR VISUALIZING AND PROCESSING PROCESS RUNS, AND A COMPUTER PROGRAM FOR SIMULATING THE SAME	)	

PRELIMINARY AMENDMENT

Assistant Commissioner for Patents
Washington, D.C. 20231

Sir:

Prior to examination of the above-captioned patent application, it is requested that the following amendments be entered.

IN THE CLAIMS:

Please replace Claims 1, 2 and 4-14 as follows.

1. (Amended) A method for visualizing and processing a value assembly process, the value assembly process being visualized as a set of value assembly lines, which value assembly lines are arranged on a number of different hierarchical levels, the value assembly lines having the following essential properties:

- each value assembly line has precisely one output interface and at least one input interface;

- the value assembly line receives input value packages via the input interfaces;

- the input value packages are combined in the value assembly line in accordance with rules defined in a specific main line function, a value contribution being made, and the value added package being generated;

- the value added package is made available via the output interface; and

- similar types of information are contained in the input value packages and in the value added package;

the process visualization having the following basic properties:

- an uppermost hierarchy level has precisely one value assembly line of the highest hierarchy level, which generates a value added package;

- the value assembly line of the highest hierarchy level receives input value packages via its input interfaces from lower-order value assembly lines;

- when a lower-order value assembly line is focused on, it is visualized in an entirely similar way as a value assembly line which receives its input value packages from lower-order value assembly lines and whose value added package is provided as input value package for the value assembly line of the uppermost hierarchy level; and

- on each hierarchy level down to a lowermost hierarchy level it is possible in each case to focus on a lower-order value assembly line of this hierarchy level, which is visualized in a similar way as a value assembly line of the next lower hierarchy level, which likewise receives input value packages via input interfaces, combines these, makes a value contribution, and makes a value added package available at the output interface;

in such a way that the process is visualized as a fractal process in the case of which the structure of all value assembly lines is similar on all hierarchy levels,

value packages being processed in accordance with the following steps:

- on a lowermost hierarchy level, value packages are supplied to the value assembly lines of the lowermost hierarchy level across the system boundaries of the value assembly process under consideration;

- the input value packages of a lowermost hierarchy level are combined in value assembly lines of this lowermost hierarchy level in accordance with its main line function, the value of the value packages is increased by a value contribution of the value assembly line and/or of the main line function, and a value added package is made available at the output interface of the value assembly line;

- on all hierarchy levels up to a highest hierarchy level, the value added package is passed on to precisely one value assembly line of the next higher hierarchy level, and serves this value assembly line as input value package;

- the value flows take place strictly in one direction, in each case from a lower hierarchy level into a higher hierarchy level, and the value assembly lines of a hierarchy level are not interconnected.

2. (Amended) The method as claimed in claim 1, in which each value added package is compared with a reference value added package, and in which impermissible

deviations of the value added package and of the reference value added package are detected and reported via a warning function.

4. (Amended) The computer program as claimed in claim 3, in which the data contained in the forms constitute input value packages and value added packages of the value assembly process.

5. (Amended) The computer program as claimed in claim 3, in which the data generated by an instruction sequence are classified according to their qualitative information content and stored in different classes of standardized data forms.

6. (Amended) The computer program as claimed in claim 3, in which during execution of the computer program or individual sequences of the computer program the generated value added packages are compared with a reference value added package in each case, which reference value added package contains specification data, and in which a report is made via a warning function in the event of impermissible specification deviations.

7. (Amended) The computer program as claimed in claim 3, in which machine-readable sequences of different value assembly subprocesses run on different computers, and data forms are transferred via long-distance data lines.

8. (Amended) A method for visualizing a value assembly process on an output unit of a computer system, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, in which value assembly process value packages are transferred in each case from a value assembly line of a lower-order hierarchy level into a value assembly line of a higher-order hierarchy level, input value packages being combined in each value assembly line, a value contribution being made, and a value added package being generated,

the computer system including, inter alia, a central processing unit and a pointing device in addition to the output unit,

in which method a value assembly line of one hierarchy level is visualized in each case on the output unit, said value assembly line being visualized as an arrow at the tip of which a value added package is transferred, and lower-order value assembly lines which are value assembly lines of a lower-order hierarchy level, are likewise visualized as arrows the tips of which are applied to the shaft of the arrow which visualizes the value assembly line of the higher-order hierarchy level,

it being possible for the user of the computer system to use the pointing device to select a lower-order value assembly line and to focus on the latter in such a way as to visualize it in the focused visualization in the same way as the value assembly line of the higher-order hierarchy level as an arrow with smaller arrows running up to it, and it being possible for the user to use the pointing device to select the arrow tip of the value assembly line,

whereupon the value assembly line of the higher-order hierarchy level is visualized in a similar way and such that the path of an integral part of the value assembly process firstly can be traced back without difficulty to its origin from a higher hierarchy level, and the contribution of this integral part to the overall value assembly process can be traced through all hierarchy levels.

9. (Amended) The method as claimed in claim 8, in which a computer program in accordance with claim 3 runs in the central processing unit of the computer system.

10. (Amended) The method as claimed in claim 8, in which the arrows of value assembly processes which make different value contributions are visualized with the aid of different colors and/or line thicknesses.

11. (Amended) The method as claimed in claim 8, in which a computer program runs in the central processing unit of the computer system, and in which the warning function prompts the computer system to display on the output unit an impermissible value deviation occurring on an arbitrary hierarchy level, and all arrows which visualize value assembly lines which are affected by the value deviation are visualized in a particularly emphatic type of visualization on the output unit in such a way that the impermissible deviation can be traced back immediately to its origin from a higher hierarchy level.

12. (Amended) A computer program which includes instruction sequences which prompt a computer to execute the method steps according to claim 8.

13. (Amended) A computer system for carrying out a program according to claim 3.

14. (Amended) A computer program product in which instruction sequences of the program are stored on a computer-readable medium in accordance with claim 3.

REMARKS

By way of the foregoing amendments to the claims, Claims 1, 2 and 4-14 have been amended to delete the multiple dependencies and reference letters. These changes have been made in accordance with 37 C.F.R. § 1.121 as amended on November 7, 2000. Marked-up versions of Claims 1, 2 and 4-14 indicating the changes accompany this Preliminary Amendment.

Early and favorable consideration with respect to this application is respectfully requested.

Should any questions arise in connection with this application, the undersigned respectfully requests that he be contacted at the number indicated below.

Respectfully submitted,

BURNS, DOANE, SWECKER & MATHIS, L.L.P.

By: _____

Robert S. Swecker
Registration No. 19,885

P. O. Box 1404
Alexandria, Virginia 22313-1404
(703) 836-6620

Date: June 15, 2001

Attachment to Preliminary Amendment dated June 15, 2001

Marked-up Claims 1, 2 and 4-14

1. (Amended) A method for visualizing and processing a value assembly process, the value assembly process being visualized as a set of value assembly lines [(VAL)], which value assembly lines are arranged on a number of different hierarchical levels, the value assembly lines having the following essential properties:

- each value assembly line has precisely one output interface and at least one input interface;

- the value assembly line receives input value packages [(IVP)] via the input interfaces;

- the input value packages are combined in the value assembly line in accordance with rules defined in a specific main line function [(MLF)], a value contribution being made, and the value added package [(VAP)] being generated;

- the value added package is made available via the output interface; and

- similar types of information are contained in the input value packages and in the value added package;

the process visualization having the following basic properties:

- an uppermost hierarchy level [(N)] has precisely one value assembly line [(VAL.N)] of the highest hierarchy level, which generates a value added package [(VAP.N)];

- the value assembly line of the highest hierarchy level receives input value packages [(IVP)] via its input interfaces from lower-order value assembly lines [(SAL.N)];

Attachment to Preliminary Amendment dated June 15, 2001

Marked-up Claims 1, 2 and 4-14

- when a lower-order value assembly line [(SAL.N)] is focused on, it is visualized in an entirely similar way as a value assembly line [(VAL.N-1)] which receives its input value packages from lower-order value assembly lines [(SAL.N-1)] and whose value added package [(VAP.N-1)] is provided as input value package for the value assembly line of the uppermost hierarchy level [(VAL.N)]; and

- on each hierarchy level [(M)] down to a lowermost hierarchy level it is possible in each case to focus on a lower-order value assembly line [(SAL.M)] of this hierarchy level, which is visualized in a similar way as a value assembly line [(VAL.M-1)] of the next lower hierarchy level [(M-1)], which likewise receives input value packages via input interfaces, combines these, makes a value contribution, and makes a value added package [(VAP.M-1)] available at the output interface;

in such a way that the process is visualized as a fractal process in the case of which the structure of all value assembly lines is similar on all hierarchy levels,

value packages being processed in accordance with the following steps:

- on a lowermost hierarchy level, value packages [(IVP)] are supplied to the value assembly lines [(VAL)] of the lowermost hierarchy level across the system boundaries of the value assembly process under consideration;

- the input value packages [(IVP)] of a lowermost hierarchy level are combined in value assembly lines [(VAL)] of this lowermost hierarchy level in accordance with its main line function [(MLF)], the value of the value packages is increased by a value contribution of the

Attachment to Preliminary Amendment dated June 15, 2001

Marked-up Claims 1, 2 and 4-14

value assembly line and/or of the main line function, and a value added package [(VAP)] is made available at the output interface of the value assembly line;

- on all hierarchy levels up to a highest hierarchy level, the value added package [(VAP)] is passed on to precisely one value assembly line [(VAL)] of the next higher hierarchy level, and serves this value assembly line as input value package;

- the value flows take place strictly in one direction, in each case from a lower hierarchy level into a higher hierarchy level, and the value assembly lines of a hierarchy level are not interconnected.

2. (Amended) The method as claimed in claim 1, in which each value added package [(VAP)] is compared with a reference value added package [(RVAP)], and in which impermissible deviations of the value added package [(VAP)] and of the reference value added package [(RVAP)] are detected and reported via a warning function.

4. (Amended) The computer program as claimed in claim 3, in which the data contained in the forms constitute input value packages [(IVP)] and value added packages [(VAP)] of the value assembly process.

Attachment to Preliminary Amendment dated June 15, 2001

Marked-up Claims 1, 2 and 4-14

5. (Amended) The computer program as claimed in [one of claims] claim 3 [or 4], in which the data generated by an instruction sequence are classified according to their qualitative information content and stored in different classes of standardized data forms.

6. (Amended) The computer program as claimed in [one of claims] claim 3 [to 5], in which during execution of the computer program or individual sequences of the computer program the generated value added packages [(VAP)] are compared with a reference value added package [(RVAP)] in each case, which reference value added package contains specification data, and in which a report is made via a warning function [(EWS)] in the event of impermissible specification deviations [(NC)].

7. (Amended) The computer program as claimed in [one of claims] claim 3 [to 6], in which machine-readable sequences of different value assembly subprocesses run on different computers, and data forms are transferred via long-distance data lines.

8. (Amended) A method for visualizing a value assembly process on an output unit of a computer system, the process comprising a number of self-similar value assembly lines which are arranged on different hierarchy levels and are independent of one another on a hierarchy level, in which value assembly process value packages are transferred in each case from a value assembly line of a lower-order hierarchy level into a value assembly line of a

Attachment to Preliminary Amendment dated June 15, 2001

Marked-up Claims 1, 2 and 4-14

higher-order hierarchy level, input value packages being combined in each value assembly line, a value contribution being made, and a value added package being generated,

the computer system including, inter alia, a central processing unit and a pointing device in addition to the output unit,

in which method a value assembly line [(VAL.M)] of one hierarchy level is visualized in each case on the output unit, said value assembly line being visualized as an arrow at the tip of which a value added package [(VAP.M)] is transferred, and lower-order value assembly lines [(SAL.M)] which are value assembly lines of a lower-order hierarchy level, are likewise visualized as arrows the tips of which are applied to the shaft of the arrow which visualizes the value assembly line of the higher-order hierarchy level,

it being possible for the user of the computer system to use the pointing device to select a lower-order value assembly line and to focus on the latter in such a way as to visualize it in the focused visualization in the same way as the value assembly line of the higher-order hierarchy level as an arrow with smaller arrows running up to it, and it being possible for the user to use the pointing device to select the arrow tip of the value assembly line, whereupon the value assembly line of the higher-order hierarchy level is visualized in a similar way and such that the path of an integral part of the value assembly process firstly can be traced back without difficulty to its origin from a higher hierarchy level, and the contribution of this integral part to the overall value assembly process can be traced through all hierarchy levels.

Attachment to Preliminary Amendment dated June 15, 2001

Marked-up Claims 1, 2 and 4-14

9. (Amended) The method as claimed in claim 8, in which a computer program in accordance with [one of claims] claim 3 [to 7] runs in the central processing unit of the computer system.

10. (Amended) The method as claimed in [one of claims] claim 8 [or 9], in which the arrows of value assembly processes which make different value contributions are visualized with the aid of different colors and/or line thicknesses.

11. (Amended) The method as claimed in [one of claims] claim 8 [to 10], in which a computer program [in accordance with claim 6] runs in the central processing unit of the computer system, and in which the warning function prompts the computer system to display on the output unit an impermissible value deviation occurring on an arbitrary hierarchy level, and all arrows which visualize value assembly lines which are affected by the value deviation are visualized in a particularly emphatic type of visualization on the output unit in such a way that the impermissible deviation can be traced back immediately to its origin from a higher hierarchy level.

12. (Amended) A computer program which includes instruction sequences which prompt a computer to execute the method steps according to [one of claims] claim 8 [to 11].

Attachment to Preliminary Amendment dated June 15, 2001

Marked-up Claims 1, 2 and 4-14

13. (Amended) A computer system for carrying out a program according to [one of claims] claim 3 [to 7].

14. (Amended) A computer program product in which instruction sequences of the program are stored on a computer-readable medium in accordance with [one of claims] claim 3 [to 7].